

PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX BERBASIS EQUIVALENCE PARTITIONS PADA APLIKASI SISTEM INFORMASI SEKOLAH

M. Komarudin MZ

Program Diploma 3 Manajemen Informatika UM. Metro

Jalan Ki. Hajar Dewantara No. 116 Iring Mulyo Kota Metro 34111

ABSTRACT

This study was conducted to help SDIT Al Qudwah Punggur Lampung in testing the accuracy of the School of Information Systems software used by the school. School Information System will be tested by the method of Black Box based Equivalensi Partition thoroughly with regard to the use, benefits, and results in find of use of the software. In software testing information systems School SDIT Al Qudwah Punggur Central Lampung consists of 12 forms. Black Box method based partitions Equivalensi test the level of accuracy will be the documentation of software testing with ditemukanya error on each form that is divided into five models of error, such as: error on function, data structures, interfaces, Initialization, and Performance. Results from this study will provide recommendations on testing the accuracy of the School of Information Systems software on SDIT Al Quqwah Punggur Central Lampung and will provide the best solution for the school for the future.

Keywords: Equivalensi Partitions, School of Information Systems, Methods Black Box.

ABSTRAK

Penelitian ini dilakukan untuk membantu SDIT Al Qudwah Punggur Lampung dalam menguji tingkat akurasi perangkat lunak Sistem Informasi Sekolah yang digunakan oleh sekolah tersebut. Sistem Informasi Sekolah akan diuji dengan Metode Black Box berbasis Equivalensi Partition secara menyeluruh berkenaan dengan penggunaan, manfaat, dan hasil yang di dapati dari pemanfaatan perangkat lunak tersebut. Dalam pengujian perangkat lunak Sistem informasi Sekolah SDIT Al Qudwah Punggur Lampung Tengah terdiri dari 12 form. Metode Black Box berbasis Equivalensi partitions menguji tingkat akurasi yang akan dilakukan dokumentasi pengujian perangkat lunak dengan ditemukanya kesalahan pada setiap form yang dibagi menjadi lima model kesalahan, diantaranya : kesalahan pada Fungsi, Struktur data, Interface, Inisialisasi, dan Performance. Hasil dari penelitian ini akan memberikan rekomendasi atas pengujian tingkat akurasi perangkat lunak Sistem Informasi Sekolah pada SDIT Al Quqwah punggur Lampung Tengah dan akan memberikan solusi terbaik bagi sekolah tersebut untuk masa yang akan datang.

Kata Kunci: Equivalensi Partitions, Sistem Informasi Sekolah, Metode Black Box.

1. PENDAHULUAN

1. 1 Latar Belakang

Pengujian adalah suatu proses pengeksekusian program yang bertujuan untuk menemukan kesalahan (Berard, 1994: p5). Pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya. Sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi (Fournier, Cs, 2009: p10).

Ada pun tujuan pengujian perangkat lunak itu sendiri yaitu untuk menemukan kesalahan yang menyebabkan perangkat lunak yang telah di bangun gagal. Selain tujuan diatas, pengujian perangkat lunak juga bertujuan untuk memperoleh produk yang berkualitas yang memberikan produktivitas tinggi (Oscar Pastor, Cs, 2007: p1). Dalam proses pengujian perangkat lunak, untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan inputan dengan output yang diinginkan (Mark Last, Cs, 2002: p1).

Pengujian otomatis harus dirancang dengan baik agar dapat menemukan klasifikasi kesalahan secara sistematis dan dapat diperbaiki dalam waktu dan usaha yang minimal (Hendrowati, 2003: p3).

Pengujian Black-Box berusaha untuk menemukan kesalahan dalam beberapa

kategori, diantaranya: fungsi-fungsi yang salah atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses database eksternal, kesalahan performa, kesalahan inisialisasi dan terminasi (B. B. Agarwad, 2010: p1).

Sejumlah peneliti melakukan penelitian pada pengujian perangkat lunak secara Black-Box . Salah satunya dilakukan oleh Mark Last, Cs pada tahun 2002 (Mark Last, Cs, 2002: p1), Patrick J, Cs, dalam penelitiannya, mereka membahas tentang pengujian reduksi dengan menggunakan input-output perangkat lunak sebagai kategori yang akan dicari kesalahannya dengan metode Black-Box.

Dalam proses pengujian perangkat lunak, untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan inputan dengan output yang diinginkan (Mark Last, Cs, 2002: p1).

Pengujian otomatis harus dirancang dengan baik agar dapat menemukan klasifikasi kesalahan secara sistematis dan dapat diperbaiki dalam waktu dan usaha yang minimal (Hendrowati, 2003: p3).

Pengujian Black-Box berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya: fungsi-fungsi yang salah atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses database eksternal, kesalahan performa, kesalahan inisialisasi dan terminasi (B. B. Agarwad, 2010: p1).

Sejumlah peneliti melakukan penelitian pada pengujian perangkat lunak secara Black-Box . Salah satunya dilakukan oleh Mark Last, Cs pada tahun 2002 (Mark Last, Cs, 2002: p1), Patrick J, Cs, dalam penelitiannya, mereka membahas tentang pengujian reduksi dengan menggunakan input-output perangkat lunak sebagai kategori yang akan dicari kesalahannya dengan metode Black-Box.

Pengujian perangkat lunak secara manual tidak efektif, Menurut studi dari Quality Assurance Institute(QAQ95A), yang menggunakan ukuran jumlah test cases dalam membandingkan pengujian perangkat lunak secara manual dan secara otomatis, dimana digunakan 1750 test cases dan ditemukan 700 errors.

Dari latar belakang diatas terbukti bahwa pengujian perangkat lunak perlu adanya, oleh dari itu dalam penelitian ini, peneliti mencoba menggunakan pendekatan metode Black-Box berbasis Equivalence Partitions, dengan harapan dalam pengujian perangkat lunak bisa lebih efektif.

1.2 Ruang Lingkup Penelitian

Masalah yang diteliti Pengujian Perangkat Lunak terhadap system informasi sekolah, dengan aspek yg diteliti faktor **fungsi-fungsi** yang salah atau hilang, kesalahan **interface**, kesalahan dalam **struktur data** atau akses *database* eksternal, kesalahan **performa**, kesalahan **inisialisasi** dan **terminasi**. Objek

penelitian dilakukan di SD IT di kabupaten Lampung Tengah.

1.3 Perumusan Masalah

Perumusan masalah dari penelitian ini: Bagaimana efektifitas pengujian perangkat lunak metode Black-Box berbasis Equivalence Partitions pada Sistem Informasi Sekolah.

1.4 Tujuan dan Manfaat Tesis

Tujuan penelitian ini antara lain sebagai berikut :

Menerapkan metode Black-Box berbasis Equivalence Partitions untuk pengujian perangkat lunak sehingga bisa lebih efektif.

Adapun manfaat yang di dapatkan dari penelitian ini antara lain:

1. Manfaat praktisi dari hasil penelitian ini adalah agar metode Black-Box berbasis Equivalence Partitions ini dapat digunakan oleh divisi perangkat lunak logic testing untuk pengujian perangkat lunak.
2. Manfaat teoritis dari hasil penelitian ini diharapkan dapat memberikan gambaran dan pemahaman penerapan metode Black-Box berbasis Equivalence Partitions pada suatu studi kasus pengujian perangkat lunak.
3. Manfaat kebijakan dari penelitian ini adalah dengan hasil yang lebih efektif dan dapat men-support manajemen untuk menentukan perangkat lunak yang baik.

2. Tinjauan Teori

2.1. Pengujian Perangkat Lunak

Pengujian Perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan maksud menemukan atau, melibatkan setiap kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu program atau sistem dan menentukan bahwa itu memenuhi hasil yang dibutuhkan perusahaan (Perry, 1990: p1).

Ada beberapa pendekatan pengujian perangkat lunak:

1. Pengujian Perangkat Lunak Metode Black-Box

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir (Perry, 1990:p100).

Hal ini juga disebut data-driven, input atau output didorong (Myers G. J., 1979: p5), atau persyaratan-based (Hetzel, 1988: p1) pengujian. Karena hanya fungsi dari modul perangkat lunak yang menjadi perhatian, pengujian *Black-Box* juga mengacu pada uji fungsional, metode pengujian menekankan pada menjalankan fungsi dan pemeriksaan inputan dan data output (William E. Howden, 1987: p7)

2. Pengujian Perangkat Lunak Metode White-Box

Berbeda dengan pengujian kotak hitam, perangkat lunak dipandang sebagai *white-box*, sebagai struktur dan aliran perangkat lunak yang diuji terlihat ke tester. Pengujian rencana yang dibuat sesuai dengan rincian pelaksanaan perangkat lunak, seperti bahasa pemrograman, logika, dan gaya. Uji kasus berasal dari struktur program. Pengujian kotak putih-disebut juga pengujian kaca-kotak, pengujian logika-driven (Myers G. J., 1979: p50) atau desain berbasis pengujian (Hetzel, 1988: p19).

2.2. Pengujian Performance Perangkat Lunak

Tidak semua sistem perangkat lunak memiliki spesifikasi terhadap kinerja secara eksplisit. Namun setiap sistem akan memiliki persyaratan kinerja implisit. Perangkat lunak tidak boleh mengambil waktu tak terbatas atau sumber daya yang tak terbatas untuk melaksanakan. "bug Performance" kadang-kadang digunakan untuk merujuk kepada orang-masalah desain dalam perangkat lunak yang menyebabkan kinerja sistem untuk menurun.

2.3. Pengujian Reliability Perangkat Lunak

Keandalan Perangkat Lunak mengacu pada probabilitas kegagalan-operasi bebas dari sistem. Hal ini terkait dengan banyak aspek perangkat lunak, termasuk proses pengujian. Langsung memperkirakan keandalan perangkat lunak dengan mengukur faktor-faktor yang terkait bisa sulit. Dipandu oleh

profil operasional, pengujian perangkat lunak (biasanya kotak hitam pengujian) dapat digunakan untuk memperoleh data kegagalan, dan model estimasi dapat lebih digunakan untuk menganalisis data untuk memperkirakan keandalan keandalan sekarang dan memprediksi masa depan.

Oleh karena itu, berdasarkan estimasi, para pengembang dapat memutuskan apakah akan merilis perangkat lunak, dan pengguna dapat memutuskan apakah akan mengadopsi dan menggunakan perangkat lunak. Risiko menggunakan perangkat lunak juga dapat dinilai berdasarkan informasi keandalan. pendukung bahwa tujuan utama pengujian harus untuk mengukur keandalan perangkat lunak diuji.

3. METODOLOGI PENELITIAN

3.1 Teknik Pengumpulan Data

Dalam melakukan penelitiannya, peneliti akan menggunakan metodologi ;

1. Penelitian Pustaka

Dengan metode ini peneliti mencari dan mengumpulkan informasi dari teknik dalam pengujian perangkat lunak.

2. Penelitian Lapangan

Dengan metode kedua ini peneliti akan langsung melakukan pengujian terhadap aplikasi sistem informasi sekolah dan

mewancarai user end dari aplikasi tersebut.

3.2 Teknik Analisis Data

1. Pengukuran Awal

Pengukuran awal dilakukan dengan mengkaji penelitian-penelitian sebelumnya dan mengkaji teori teori yang relevan untuk mendapatkan item item yang membentuk konstruk yang diinginkan, yaitu konstruk kegunaan persepsian (*perceived usefulness*) dan konstruk kemudahan penggunaan (*perceived ease of use*).

2. Sebelum Tes (pretest)

Sebelum tes dilakukan dalam bentuk studi pilot (*pilot study*) melibatkan beberapa partisipan untuk mengkonfirmasi item item yang membentuk konstruk konstruk tersebut. Hasil *pretest* ini untuk membuang atau mengganti item item yang kurang cocok dalam membentuk konstruk konstruk yang diinginkan.

3. Validasi Konstruk

Setelah item item ditentukan untuk membentuk konstruk konstruk, maka item item tersebut perlu diuji validitasnya. Validitas konstruk digunakan untuk menentukan item item apa saja yang valid membentuk konstruk konstruk tersebut.

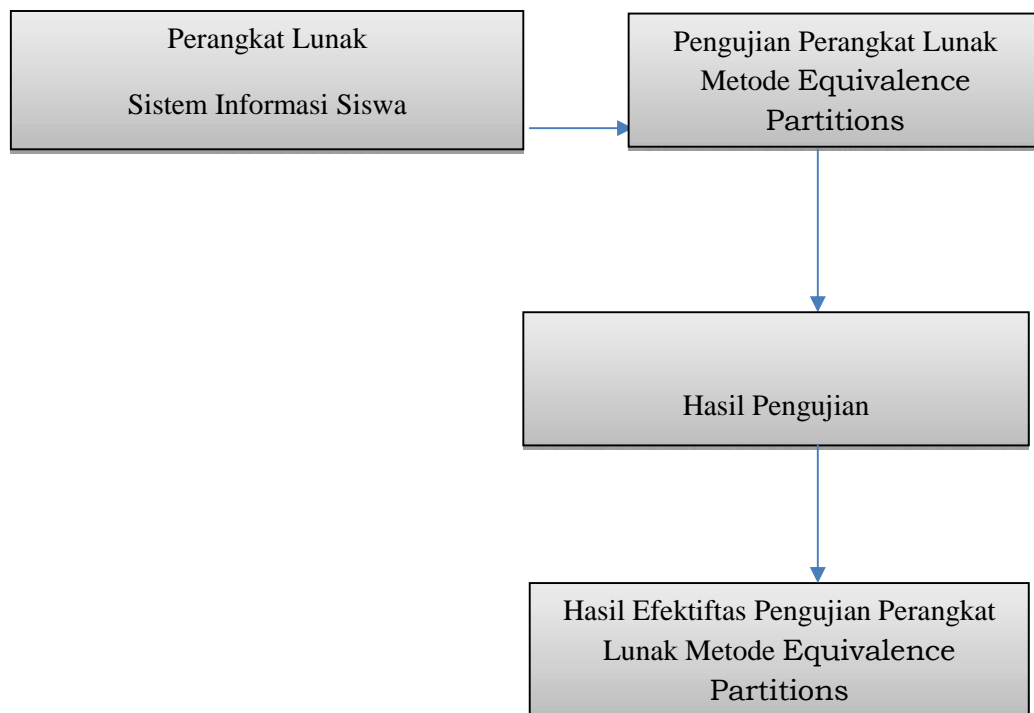
4. Reliabilitas

Setelah ditemukan item item valid untuk masing masing konstruk, maka langkah berikutnya adalah menguji reabilitasnya dengan menggunakan nilai Cronbach Alpa.

3.2 Analisa Kebutuhan

Penelitian ini menggunakan metode eksperimen, dengan tahapan sebagai berikut: perancangan penelitian, teknik analisis. Penelitian ini akan membuat suatu model prediksi tingkat efektifitas pengujian perangkat lunak metode Equivalence

Partitions, yaitu: dengan terlebih dahulu *test case* akan diuji dengan menggunakan metode Equivalence Partitions, hal ini dilakukan untuk mendapatkan dataset, kemudian dataset ini akan dihitung efektifitasnya. Berikut yang menjadi metodologi dalam penelitian ini adalah dapat dilihat pada gambar 3.1., yaitu:

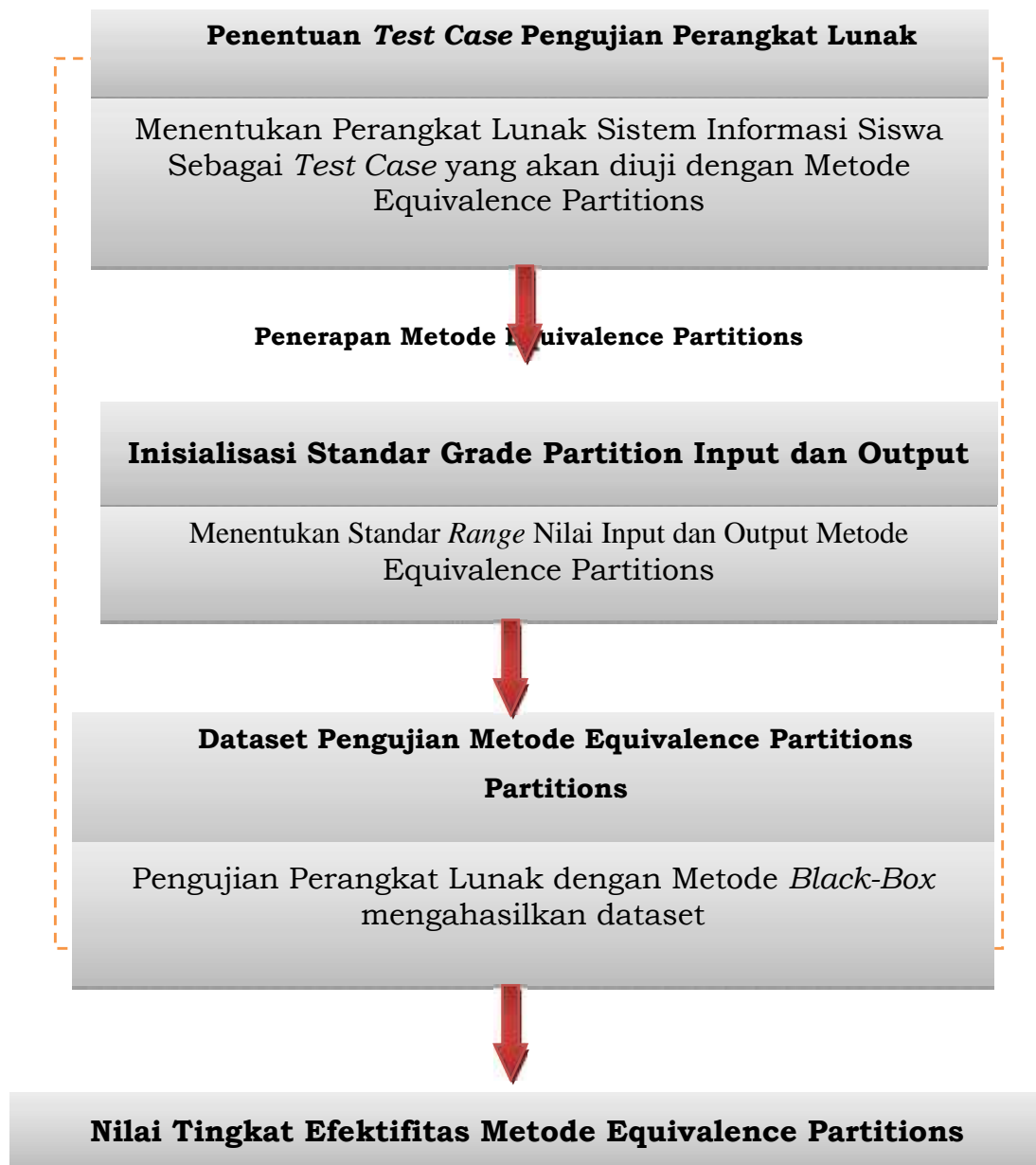


Gambar 3.1 Metodologi penelitian model pengujian perangkat lunak prediksitingkat Efektifitas berbasis Metode Equivalence Partitions.

3.3 Perancangan Penelitian

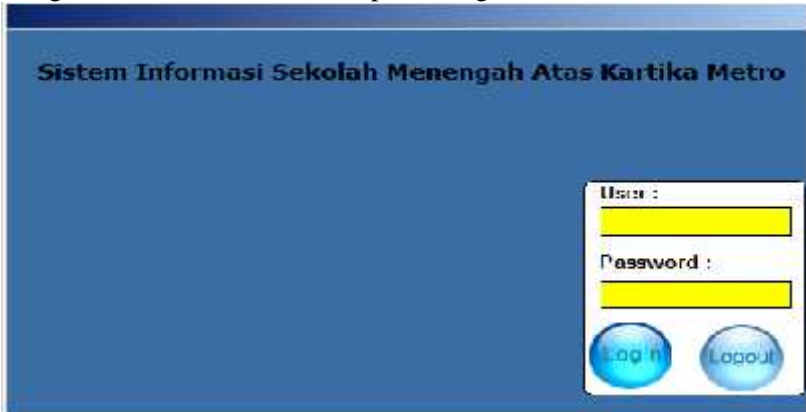
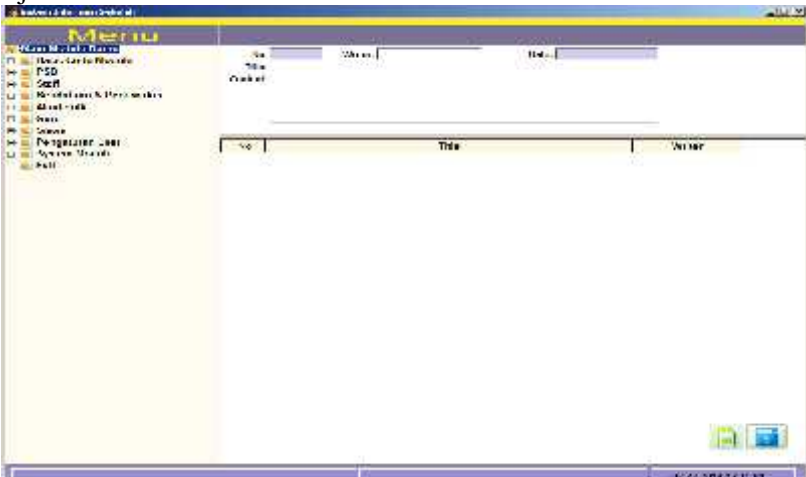
Dalam penelitian ini akan dilakukan beberapa tahapan, dimana tahapan ini dapat dilihat pada gambar 3.2, Pada tahapan pertama diawali dengan Penentuan Test Case Perangkat Lunak, Inisialisasi

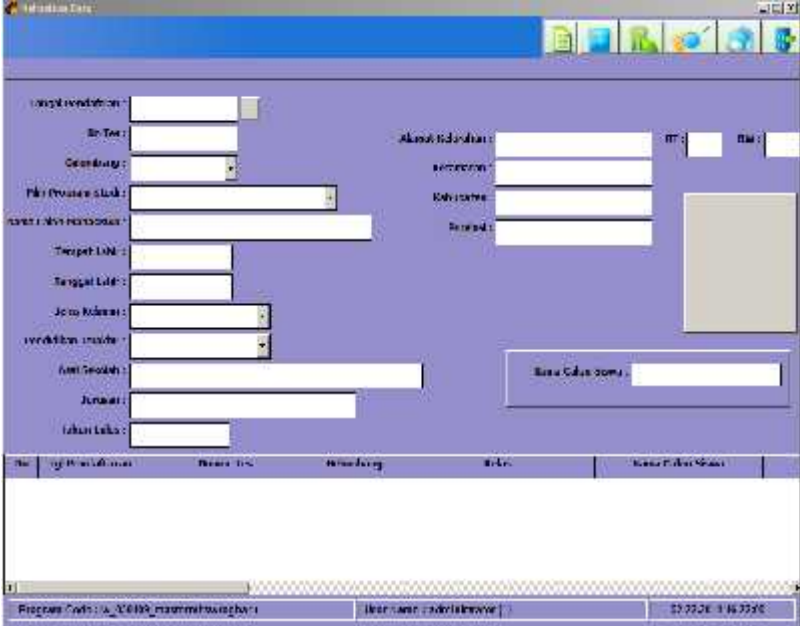
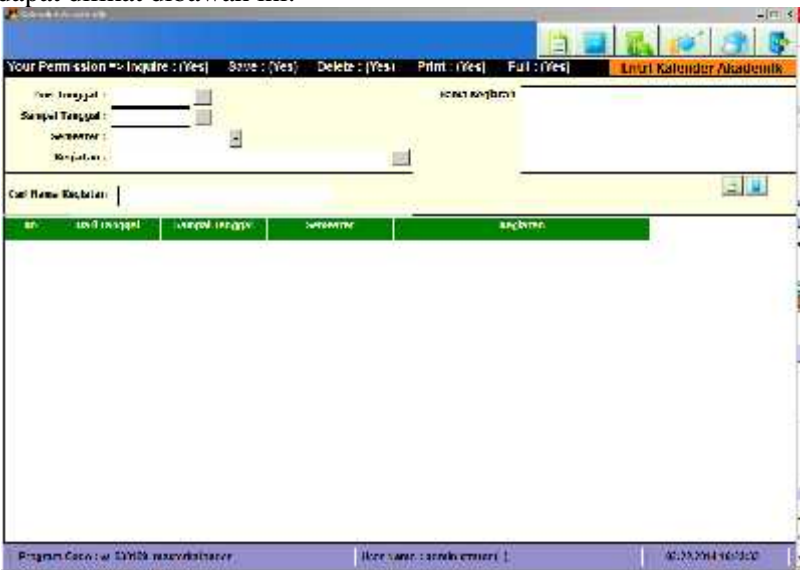
Standar Grade Partition Input Dan Output, Dokumentasi Pengujian Dengan Metode Equivalence Partitions, Nilai Tingkat Efektifitas Metode Equivalence Partitions.



Gambar 3.2 Tahapan-tahapan model pengujian perangkat

Tabel 3.1 Form Test Case Perangkat Lunak
Sistem Informasi Sekolah

| Nama Form | Deskripsi Form |
|-----------------------------------|--|
| Form Login | <p>Form Login digunakan untuk memasuki system baik itu level sebagai, Staf, Guru, Siswa, maupun sebagai Administrator.</p>  |
| Form Menu Utama | <p>Form Menu Utama digunakan untuk tampilan atau control menu aplikasi yang ada, dengan cara klik salah satu form yang yang ingin dijalankan.</p>  |
| Form Master Mata Pelajaran | <p>Form Master Mata Pelajaran digunakan untuk memasukkan data Pelajaran sekaligus query data matakuliah. Tampilan form Mata Pelajaran dapat dilihat dibawah ini.</p> |

| | |
|---|--|
| |  |
| <p>Form Master Kalender Akademik</p> | <p>Form Master Kalender Akademik digunakan untuk memasukkan data kalender akademik query data akademik. Tampilan formnya dapat dilihat dibawah ini.</p>  |
| <p>Form Transaksi Penyebaran Kelas</p> | <p>Form Transaksi Penyebaran Kelas digunakan untuk memasukkan data sebaran kelas Siswa sekaligus query data sebaran kelas Siswa. Tampilan formnya dapat dilihat di bawah ini.</p> |

Form Transaksi Entry Nilai digunakan untuk memasukkan data nilai Siswa sekaligus query nilai Siswa. Tampilan formnya dapat dilihat dibawah ini.

3.4 Score Grade Partition Pengujian Perangkat Lunak pada Error Fungsi

Pada pengujian perangkat lunak metode *Black-Box* untuk kesalahan pada Fungsi dapat dilihat pada tabel 3.2, yaitu:

Tabel 3.2. *Score Grade Partition* Pengujian Perangkat Lunak
Metode *Black-Box* pada Error Fungsi

| Form | Nama Form | Uraian Error Pada Fungsi | Jumlah Error (Kali) | Score Error |
|------|----------------------------|--|---------------------|-------------|
| 1 | Form Login | Form Login digunakan untuk memasuki system baik itu level sebagai, Staf, Guru, Mahasiswa, maupun sebagai Administrator, tapi untuk level Staff dan Guru tidak sesuai . | 1 | 10 |
| 2 | Form Menu Utama | Form Menu Utama digunakan untuk tampilan atau control menu aplikasi yang ada, dengan cara klik salah satu form yang ingin dijalankan, hasil pengujian sesuai dengan fungsinya | 0 | 0 |
| 3 | Form Master Pelajaran | Form Master Mata Kuliah sesuai dengan fungsinya untuk memasukkan data matakuliah per Jurusan, untuk Jurusan IPA, IPS, dan TKJ. | 0 | 0 |
| 4 | Form Master Guru dan Staff | Form Master Guru dan Staff sesuai dengan fungsinya untuk memasukkan data Guru dan Staff sekaligus query data Guru dan Staff, dan Operator. | 0 | 0 |
| 5 | Form Master Siswa | Form Master Siswa digunakan untuk memasukkan data Siswa sekaligus query data Siswa Pada saat pemilihan jurusan pada system untuk jurusan | 1 | 10 |

| | | | | |
|----|------------------------------------|---|---|----|
| | | IPA, IPS, dan TKJ tidak sesuai , yang sesuai hanya jurusan IPA. | | |
| 6 | Form Master Pendaftaran Siswa Baru | Form Master Pendaftaran Siswa Baru digunakan untuk memasukkan data Siswa baru dan sekaligus query data Siswa Baru. Pada saat pemilihan Jurusan pada system untuk jurusan IPA, IPS, dan TKJ tidak sesuai , yang sesuai hanya jurusan IPA. | 1 | 10 |
| 7 | Form Pemberian NIS | Form Pemberian NIS Siswa Baru digunakan untuk pemeberian NIS Siswa baru dan sekaligus query data Siswa Baru. Pada saat pemberian NIS untuk pemilihan Jurusan pada system untuk jurusan IPA, IPS, dan TKJ tidak sesuai , yang sesuai hanya Jurusan IPA. | 1 | 10 |
| 8 | Form Konfigurasi NIS | Form Konfigurasi NIS digunakan untuk Konfigurasi NIS bersdasarkan Per-Tahun. Pada saat pemilihan Jurusan pada system untuk jurusan IPA, IPS, dan TKJ tidak sesuai , yang sesuai hanya Jurusan IPA. | 1 | 10 |
| 9 | Form Master Kalender Akademik | Form Master Kalender Akademik sesuai dengan fungsinya untuk memasukkan data kalender akademik query data akademik. | 0 | 0 |
| 10 | Form Transaksi Penyebaran Kelas | Form Transaksi Penyebaran Kelas digunakan untuk memasukkan data sebaran kelas Siswa sekaligus query data sebaran kelas Siswa. | 1 | 10 |

| | | | | |
|----|--------------------------------|--|---|----|
| | | Pada saat pemilihan waktu Sekolah yang tersedia, yaitu pagi, siang, sore, sabtu & minggu tidak sesuai , yang sesuai hanya pilihan jum'at & sabtu & minggu. | | |
| 11 | Form Transaksi Entry Nilai | Form Transaksi Entry Nilai digunakan untuk memasukkan data nilai Siswa sekaligus query nilai Siswa. Pada saat pemilihan kelas pada system untuk jurusan IPA, IPS, dan TKJ tidak sesuai , yang sesuai hanya Jurusan IPA. | 1 | 10 |
| 12 | Form Master Tanda Tangan Nilai | Form Master Tanda Tangan Nilai digunakan untuk meng-memasukan data Staff yang akan mendatangi Nilai. Fungsinya tidak sesuai untuk mendatangi Nilai, justru sesuai dengan mendatangi Raport Nilai Siswa | 1 | 10 |

Jumlah Form yang diuji 12 form dengan setiap form diuji 5 kali, jadi total banyak pengujian ada 60 kali, jumlah form yang ditemukan error sebanyak 36 form dan 24 form tidak ditemukan error. Berdasarkan pengujian form pada

perangkat lunak tersebut, pada tabel 3.3 disajikan daftar solusi praktis yang dapat digunakan untuk memperbaiki sistem yang digunakan pada perangkat lunak Sistem Informasi Sekolah SDIT Al Qudwah Punggur Lampung Tengah.

Tabel 3.3 Rancangan Solusi Praktis Perbaikan Perangkat Lunak Sistem Informasi Sekolah SDIT Al Qudwah Punggur Lampung Tengah

| Celah Eror | Deskripsi | Solusi |
|---|--|---|
| Form Login, Form Menu Utama, Form Master Pelajaran, Form Master Guru dan Staf, Form Master Siswa, Form Master Pendaftaran Siswa Baru, Form Pemberian NIS, Form Konfigurasi NIS, Form Master Kalender Akademik, Akademik, Form Transaksi Penyebaran Kelas, Form Transaksi Entry Nilai, Form Master Tanda Tangan Nilai. | Jika terjadi kesalahan pada Fungsi, Struktur Data, Interface, Inisialisai dan Performance. | <ul style="list-style-type: none">- Lakukan upgrade Perangkat Lunak, jika versi terbaru sudah diciptakan.- Konfigurasi Input – output proses data yang sesuai dengan kebutuhan Sistem Informasi Sekolah. |

4. KESIMPULAN

1. Terdapat 36 jenis celah eror pada beberapa form perangkat lunak *system informasi sekolah* di SDIT Al Qudwah Punggur Lampung tengah, kesalahan pada pada Fungsi, Struktur Data, Interface, Inisialisai dan Performance dari perangkat lunak tersebut.

2. Solusi praktis peningkatan akurasi perlu dilakukan segera guna memperbaiki celah eror yang telah ditemukan, selanjutnya dilakukan pengujian keamanan secara intensif melalui jaringan internal (*whitebox penetration testing*) secara berkala oleh System Administrator atau Pengelola Sistem Informasi di sekolah, khususnya bagi yang mengelola perangkat lunak tersebut.

3. Metode *Blackbox berbasis Equivalence Partitions* merupakan solusi alternatif bagi Sekolah untuk melakukan pengujian tingkat akurasi sistem informasi sekolah, khususnya layanan *informasi sekolah*.
4. Untuk mencapai tingkat akurasi, dimana semua parameter akurasi yang terkait aspek kerahasiaan, integritas data, dan ketersediaan data dapat terpenuhi, maka harus dipertimbangkan metode lain yang dapat dijadikan tolak ukur standar keamanan informasi di lingkungan sekolah, selain menggunakan metode *Equivalence Partitions*.
5. Setiap institusi sekolah memiliki karakteristik organisasi, arsitektur dan topologi

perangkat lunak, pola pengembangan aplikasi, serta regulasi internal yang berbeda-beda antar institusi. Sehingga tidak ada cara yang spesifik dalam menggunakan parameter akurasi, solusi permasalahan akurasi, serta prosedur pengujian akurasi sistem informasi yang ada di SDIT Al Qudwah Punggur Lampung Tengah.

Kesimpulan ini menjawab hipotesis penelitian, yang menunjukkan bahwa terdapat celah eror pada beberapa form perangkat lunak sistem informasi sekolah yang ada di SDIT Al Qudwah Punggur Lampung Tengah.

REFERENSI

- [1] Agus Supriatna Somantri, M. d. (2009). Identifikasi Mutu Fisik Jagung Dengan Menggunakan Pengolahan Citra Digital Dan Jaringan Syaraf Tiruan.
- [2] Albert Endres, Cs. (2003). *Hanbook software and System Engineering, Empirical Observations, Laws and Theories*.
- [3] B. B. Agarwad, C. (2010). *Software Engineering & Testing*. Boston.
- [4] Beizer, B. (1990). *Software Testing Techniques*.
- [5] Berard, C. (1994). *Issues in the Testing of Object-Oriented Software*.
- [6] Fournier, Cs. (2009). *Essential Software Testing A Use-Case Approach*.
- [7] Hendrowati (2003). Perancangan pengujian perangkat lunak Berorientasi obyek: berbasis

- status (state-based Testing).
Jurnal Universitas Paramadina
 ,Vol. 2 No. 2.
- [8] Hetzel, W. C. (1988). *The Complete Guide to Software Testing, 2nd ed.*
- [9] Wellesley, Mass.: Wellesley, Mass. : QED Information Sciences, 1988. ISBN: 0894352423. Physical description: ix, 280 p. : ill ; 24 cm.
- [10] J.M, Z. (1992). *Introduction to artificial neural systems.*
- [11] Mark Last, Cs. (2002). Effective Black-Box Testing with Genetic Algorithms. *ACM* , 1.
- [12] Myers, G. J. (1979). *The art of software testing*. New York: New York :
- [13] Wiley, c1979. ISBN: 0471043281 Physical description: xi, 177 p. : ill. ; 24 cm.
- [14] Myers, G. (1979). *The Art of Software Testing.*
- [15] Oscar Pastor, Cs. (2007). *Model-Driven Architecture in Practice, A Software Production Environment Based on Conceptual Modeling.*
- [16] Patrick J, C. (2000). Black-Box Test Reduction Using Input-Output Analysis. *ACM* .
- [17] Perry, W. E. (1990). *A standard for testing application software*. 1990.
- S, H. (1999:p20). *Neural networks – A comprehensive Foundation.*
- [18] Siang. (2005). *Jaringan Syaraf Tiruan dan Pemogramannya.*
- [19] Sri Kusumadewi, C. (2010). *Neuro-Fuzzy*. Yogyakarta.
- [20] Sthamer, C. (1995). The Automatic Generation of Software Test Data Using Genetic Algorithms.
- [21] William E. Howden, C. (1987). *Functional program Testing and Analysis.*